AFRL-VA-WP-TP-2002-315

TASK ALLOCATION FOR WIDE AREA SEARCH MUNITIONS VIA ITERATIVE NETWORK FLOW

Corey Schumacher
Phillip R. Chandler
Steven R. Rasmussen

MAY 2002

Approved for public release; distribution is unlimited.

AIR VEHICLES DIRECTORATE
AIR FORCE RESEARCH LABORATORY
AIR FORCE MATERIEL COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
May 2002	Conference Paper Preprint	

4. TITLE AND SUBTITLE
TASK ALLOCATION FOR WIDE AREA SEARCH MUNITIONS VIA ITERATIVE NETWORK FLOW

5a. CONTRACT NUMBER IN-HOUSE
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER N/A
5d. PROJECT NUMBER N/A
5e. TASK NUMBER N/A
5f. WORK UNIT NUMBER N/A

6. AUTHOR(S)
Corey Schumacher
Phillip R. Chandler
Steven R. Rasmussen

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
The Control Theory Optimization Branch (VACA)
Control Sciences Division
Air Vehicles Directorate
Air Force Research Laboratory, Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

8. PERFORMING ORGANIZATION REPORT NUMBER
AFRL-VA-WP-TP-2002-315

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Vehicles Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson Air Force Base, OH 45433-7542

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/VACA

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-VA-WP-TP-2002-315

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
Presented at the AIAA Guidance Navigation and Control Conference, Monterey, CA, August 5-8, 2002.
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT *(Maximum 200 Words)*
This paper addresses the problem of task allocation for wide area search munitions. The munitions are required to search for, classify, attack, and verify the destruction of potential targets. It is assumed that target field information is communicated between all elements of the swarm. A network flow optimization model is used to develop a linear program for optimal resource allocation. This method can be used to generate a tour of several assignments to be performed consecutively, by running the assignment iteratively and only updating the assigned task with the shortest ETA in each iteration. Periodically re-solving the overall optimization problem results in coordinated action by the search munitions. Simulation results are presented for a swarm of eight vehicles searching an area containing three potential targets. All targets are quickly serviced without using up an excessive amount of potential search time.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT:	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON (Monitor)
Unclassified	Unclassified	Unclassified	SAR	14	Corey Schumacher **19b. TELEPHONE NUMBER** *(Include Area Code)* (937) 255-8678

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

TASK ALLOCATION FOR WIDE AREA SEARCH MUNITIONS VIA ITERATIVE NETWORK FLOW

Corey Schumacher, Phillip R. Chandler
Flight Control Division
Air Force Research Laboratory (AFRL/VACA)
Wright-Patterson AFB, OH 45433-7531

Steven R. Rasmussen
Veridian Inc.
Wright-Patterson AFB, OH

Abstract

This paper addresses the problem of task allocation for wide area search munitions. The munitions are required to search for, classify, attack, and verify the destruction of potential targets. It is assumed that target field information is communicated between all elements of the swarm. A network flow optimization model is used to develop a linear program for optimal resource allocation. This method can be used to generate a "tour" of several assignments to be performed consecutively, by running the assignment iteratively and only updating the assigned task with the shortest ETA in each iteration. Periodically re-solving the overall optimization problem results in coordinated action by the search munitions. Simulation results are presented for a swarm of eight vehicles searching an area containing three potential targets. All targets are quickly serviced without using up an excessive amount of potential search time.

Introduction

Autonomous wide area search munitions (WASM) are small, powered air vehicles, each with a turbojet engine and sufficient fuel to fly for a short period of time. They are deployed in groups, or "swarms," from larger aircraft flying at higher altitudes. They are individually capable of searching for, recognizing, and attacking targets. Cooperation between munitions has the potential to greatly improve their effectiveness in many situations. The ability to communicate target information to one another will greatly improve the capability of future search munitions.

In this paper we describe a time-phased network optimization model designed to perform task allocation for a group of powered munitions each time it is run. The model is run simultaneously and independently on all munitions at discrete points in time, and assigns each vehicle a task each time it is run. The model is solved each time new information is brought into the system, typically because a new target has been discovered or an already-known target's status has been changed. A network model for task allocation was studied in [6], but that work has some limitations. One limitation of the work in [6] is that only one vehicle can be assigned to each target at a time. This is inefficient, because it does not make use of all available information. A single task is given to each vehicle, not taking into account the succeeding tasks that will need to be performed. Each target, in [6], is only responsible for a single task in the assignment at any time. In the present work, the network optimization model is run iteratively so that all of the

known targets will be completely serviced by the resulting allocation. Classification, attack, and battle damage assessment tasks can all be assigned to different vehicles when a target is found, resulting in the target being more quickly serviced. A single vehicle can also be given multiple task assignments to be performed in succession, if that is more efficient than having multiple vehicles perform the tasks individually.

The cooperative control algorithm is being implemented in a simulation with up to ten wide area search munitions and ten potential targets. This simulation has six degree-of-freedom dynamics for the search munitions and the capability to include a variety of target types. This paper presents simulation results for a swarm of vehicles searching an area containing a cluster of targets. The vehicles have limited flight times due to fuel constraints, and have an ATR capability. The vehicles are assumed to be able to communicate target state information to each other, as well as the calculated "benefits" for each vehicle performing each possible task.

Scenario

We begin with a set of N vehicles, deployed simultaneously, each with a life span of 30 minutes. We index them $i = 1, 2, \ldots, N$. Targets that might be found by searching fall into known classes according to the value or "score" associated with destroying them. We index them with j as they are found, so that $j = 1, 2, \ldots M$ and Vj is the value of target j. We assume that at the outset there is no precise information available about the number of targets and their locations. This information can only be obtained by the vehicles carrying out searches and finding potential targets using Automatic Target Recognition (ATR) methodologies. The ATR process is modeled using a system that provides a probability that the target has been correctly classified. The probability of a successful classification is based on the viewing angle of the vehicle relative to the target. At this time, the possibility of incorrect identification is not modeled, but targets are not attacked unless a 90% probability of correct identification is achieved. Further details of the ATR methodology can be found in [2], and a detailed discussion is available in [3].

Network Optimization Model

Network optimization models are typically described in terms of supplies and demands for a commodity, nodes that model transfer points, and arcs that interconnect the nodes and along which flow can take place. To model weapon system allocation, we treat the individual vehicles as discrete supplies of single units, tasks being carried out as flows on arcs through the network, and ultimate disposition of the vehicles as demands. Thus, the flows are 0 or 1. We assume that each vehicle operates independently, and makes decisions when new information is received. These decisions are determined by the solution of the network optimization model. The receipt of new target information triggers the formulation and solving of a fresh optimization problem that reflects current conditions, thus achieving feedback action. At any point in time, the database onboard each vehicle contains a *target* set, consisting of indexes, types and locations for targets that have been classified above the probability threshold. There is also a *speculative* set, consisting of indexes, types and locations for potential targets that have been detected, but are classified below the probability threshold and thus require an additional look before striking. Figure 1 provides an illustration of this model.

The model is demand driven, with the large rectangular node on the right exerting a demand-pull of N units (labeled with a supply of –N), so that each of the LOCAAS nodes on the left (with supply of +1 unit each) must flow through the network to meet the demand. In the middle layer, the top M nodes represent all of the targets that have been identified with the required minimum classification probability at this point in time and thus are ready to be attacked. An arc exists from a specific vehicle node to a target node if and only if it is a feasible vehicle/target pair. At a minimum, the feasibility requirement would mean that there is enough fuel remaining to strike the target if tasked to do so. Other feasibility conditions could also enter in, if, for example, there were differences in the onboard weapons that precluded certain vehicle/target combinations, or if the available attack angles were unsuitable. The bottom R nodes of the middle layer represent all of the potential targets that have been identified, but do not meet the minimum classification probability. We call them *speculatives*. The minimum feasibility requirement for an arc to connect a vehicle /speculative pair is sufficient fuel for the vehicle unit to assume a position in which it can deploy its sensor to assist in elevating the classification probability beyond threshold. The lower tier models alternatives for battle damage assessment for targets that have been struck. Finally, each node in the vehicle set on the left has a direct arc to the far right node labeled sink, modeling the option of continuing to search. The capacities on the arcs from the target and speculative sets are fixed at 1. Due to the integrality property, the flow values are constrained to be either 0 or 1. Each unit of flow along an arc has a "benefit" which is an expected future value. The optimal solution maximizes total value.

The network optimization model can be expressed as:

$$\max J = \sum_{i,j} c_{ij} x_{ij} \qquad (1)$$

Subject to:

$$\sum_{i,j} x_{ij} + x_{jk} = 1, \qquad ,\forall i = 1,\ldots,n \qquad (2)$$

$$\sum_{i} x_{is} + \sum_{j} x_{jk} = n, \qquad ,n = \#UAVs \qquad (3)$$

$$x \geq 0 \qquad (4)$$

$$x_{is} \leq 1 \qquad (5).$$

This particular model is a capacitated transshipment problem (CTP), a special case of a linear programming problem. Constraint (2) enforces a condition that flow-in must equal flow-out for all nodes. Constraint (3) forces the number of assigned tasks to be equal to the number of available vehicles. Constraints (4) and (5) help enforce the binary nature of the problem. Any particular flow is either active or inactive (0 or 1). Restricting these capacities to a value of one on the arcs leading to the sink, along with the integrality property, induces binary values for the decision variables x_{ij}. Due to the special structure of the problem, there will always be an optimal solution that is all integer [1]. Solutions to this problem pose a small computational burden, making it feasible for implementation on the processors likely to be available on disposable wide area search munitions.

The goal of the optimization problem is to maximize the value of the tasks performed by the vehicles at the time the model is solved. Solving the model whenever new target information is available attempts to maximize the value of the targets destroyed over the life of the munitions.

Due to the integrality property, it is not normally possible to simultaneously assign multiple vehicles to a single target, or multiple targets to a single vehicle. However, using the network assignment iteratively, "tours" of multiple assignments can be determined. This is done by solving the initial assignment problem once, and only finalizing the assignment with the shortest ETA. The assignment problem can then be updated assuming that assignment is performed, updating target and vehicle states, and running the assignment again. This iteration can be repeated until all of the vehicles have been assigned terminal attack tasks, or until all of the target assignments have been fully distributed. The target assignments are complete when classification, attack, and battle damage assessment tasks have been assigned for all known targets. One limitation of this method is that the assignments must be recomputed if a new target is found or a munition fails to complete an assigned task.

Another complication arises from the attempted decoupling between path planning and task assignment. Potential trajectories are calculated for each vehicle performing each needed task, and these are then sent to the assignment algorithm and used in calculating the task benefits cij. However, the timing constraints between tasks are not considered in calculating these trajectories. When the benefits are calculated, however, a benefit of zero is given to any path that would result in a task being performed before the necessary prerequisite tasks. For example, the benefit for any verification task is zero if the corresponding path would result in verification being performed before the target has been attacked. If an attack path is chosen that takes longer than any of the pre-calculated verification paths, then verification will not be assigned. This occurs in the simulation example to follow. The verification task could be assigned at a later time if the assignment algorithm was run again, with different initial conditions. At the worst, the assignment algorithm could be run when the final target was attacked, and it would then be guaranteed that all of the available munitions would calculate verification trajectories that met all the timing constraints, since the target has already been attacked.

Simulation

This network flow model has been implemented in our multi-vehicle, multi-target coordinated-control simulation. The scenario has eight Wide Area Search Munitions performing a search for targets in a rectangular area. The WASM are using a simple "moving the grass" search pattern. There are up to 5 different target types possible in the simulation, including a "non-target" target type for objects that appear similar to targets but which may be distinguishable as non-targets by the ATR.

One of the critical questions involved in using the network flow model for coordinated control and decision-making for WASM is how the values of the weights $c(i,j)$ are chosen. Different values will achieve good results for different situations. For example, reduced warhead effectiveness greatly increases the importance of battle damage assessment and potential repeated attacks on an individual target. A simplified scheme has been developed which does

not attempt to address the full probabilistic computation of the various Expected Values. It is intended to assign the highest value possible to killing a target of the highest-valued type, with other tasks generating less of a benefit. The values of different tasks are calculated as follows:

C(i,j) =Expected value of vehicle I attacking target j
= (Probability target type has been correctly identified)*(Probability of destroying target j) * (Value of target j)*(Time weighting)*(Previous task weighting)
$= P_{id} * P_k * V_j * \min_j(\text{ETAMatrix})/\text{ETAMatrix}(i,j)$

C(i,s) = Value of vehicle i continuing to search
= (Maximum Target Value)*(Remaining flight time)/(Maximum flight time)
$= \max(\text{target values}) * T_f/T_m$

C(i,k) = Expected value of vehicle i assisting in classifying speculative k
= ((Probability successful ATR)*(Expected value of target being attacked after classification) + Value of continued search after classification) *(Previous task weighting)
$= (Patr*Pk*Vj + \max(\text{target values})*(T_f - T_{classify})/T_m)$

C(i,g) = Expected value of vehicle i performing BDA on target g
= ((Probability successful BDA)*(Probability target was not killed)(Probability of correct target ID)(Value of target j) + Value of continued search after classification) *(Previous task weighting)
$= (P_{bda}*(1-P_k)*P_{id}*V_j + \max(\text{target values}) *(T_f - T_{bda})/T_{mg})$

There are five possible target types with different values, and different ATR characteristics. P_{id} is an input based on the quality of the ATR recognition. ETAMatrix contains the required flight times for each vehicle i to fly to each target j. T_f is the remaining available flight time of a vehicle, and T_m is the maximum flight time of the vehicle. For the following simulation results, some of the parameters were set as constants: $P_k = 0.80$, $P_{bda} = 1.0$. $T_{classify}$ and T_{bda} are equal to the flight time to reach the specified target, plus the time needed to return to search after the task is completed.

The value of attacking a target is weighted with the time required for a vehicle to perform that attack, so that a higher value is assigned to a vehicle that can attack a target sooner. The value of continuing to search is set such that the value of searching is equal to the value of killing a high-value target initially, and degrades linearly with search time remaining. This will tend to result in vehicles with less flight time remaining being used to kill targets, and vehicles with more fuel left being used to search, classify, and perform BDA. Determining precise appropriate values for the probabilities of successful ATR and BDA is difficult, and requires substantial modeling of those processes, which this paper does not address in substantial detail. Simplified models giving reasonable values for these parameters are used. The value of all possible tasks, vehicle, and target assignment combinations are calculated and sent to the capacitated transshipment problem solver. The values are multiplied by 1,000 before being sent to the solver, as it only works with integers and rounding will result in poor results without the scaling factor.

For the simulation results presented, eight vehicles are searching an area containing three targets of different types, and hence of different values. The target information is as follows:

Target	Type	Value	Location (X,Y)
1	1	10	(5000,-1500)
2	3	8	(6500,-500)
3	2	10	(11000,-2500)

The targets also have an orientation (facing) that has an impact on the ATR process and desired viewing angles, but this will not be discussed as it does not directly affect the task allocation. The search vehicles are initialized in a staggered row formation, with fifteen minutes of flight time remaining, out of a maximum thirty minutes. This assumes that the vehicles have been searching for fifteen minutes and then find a cluster of potential targets. Figure 2 shows simulation results with the iterative computation of tour assignments. The colored rectangles represent the sensor footprints of the searching vehicles, and the numbers are the target locations. Colored lines show flight paths. Targets are numbered 1-3. As soon as each target is discovered, classification, attack, and possibly verification (if time constraints allow) tasks are assigned for that target. Since the task allocation algorithm is performed each time a task is completed, if it possible for a vehicle's assignment to change based on new target information. There is one instance where a vehicle (#2) is pulled off its search path to perform another task, and then reassigned to search before completing that task. This churning could be reduced with a small "memory weighting" encouraging vehicles to perform tasks which they'd already been assigned. All of the targets are fully serviced (found, classified, attacked, and verified as destroyed) in this example, except for Target 1. No slow enough verification path has been calculated at the time of the last assignment, due to the multiple tasks and long path assigned to Vehicle 6 before the attack is performed. Verification could be assigned with a later assignment computation, and this will be done in future work.

V. Conclusions

In this paper we presented a solution to the problem of task allocation for wide area search munitions. The vehicles are capable of searching for targets, performing ATR to classify targets, attack targets, and perform BDA on targets. A linear program based on the capacitated trans-shipment problem is used to solve the task allocation problem. Simulation results are presented for eight vehicles searching and attacking three targets of different values within the search area. The network optimization results in an effective allocation of vehicle resources to the required tasks. Results for an iterative implementation of the network flow algorithm are given. This method allows assignment of multiple vehicles to a single target, and multiple targets to a single vehicle. The resulting assignment is sub-optimal, but is effective, and can be implemented in real-time with relatively low computational requirements. Methods and metrics for comparing different sub-optimal assignment methodologies need to be developed.

VI. References

1. Nygard, K. E., Chandler, P R. , Pachter, M.., "Dynamic Network Flow Optimization Models for Air Vehicle Resource Allocation," Proceedings of the Automatic Control Conference, June 2001.
2. Chandler, Phillip R., Pachter, Meir, ``UAV Cooperative Classification", to appear in Workshop on Cooperative Control and Optimization, Klewer Academic Publishers, 2001.
3. Chandler, Phillip R., Pachter, Meir, "Hierarchical Control of Autonomous Teams," Proceedings of the 2001 AIAA Guidance, Navigation, and Control Conference
4. Ford, L. R Jr. and D. R. Fulkerson, "Flows in Networks," Princeton University Press, Princeton, NJ, 1962
5. Murphy, R.A., "An Approximate Algorithm For a Weapon Target Assignment Stochastic Program," Approximation and Complexity in Numerical Optimization: Continuous and Discrete Problems, editor: P.M. Pardalso
6. Schumacher, C, Chandler, P.R, Rasmussen, S. R., "Task Allocation for Wide Area Search Munitions Via Network Flow Optimization", Proceedings of the 2001 AIAA Guidance, Navigation, and Control Conference.

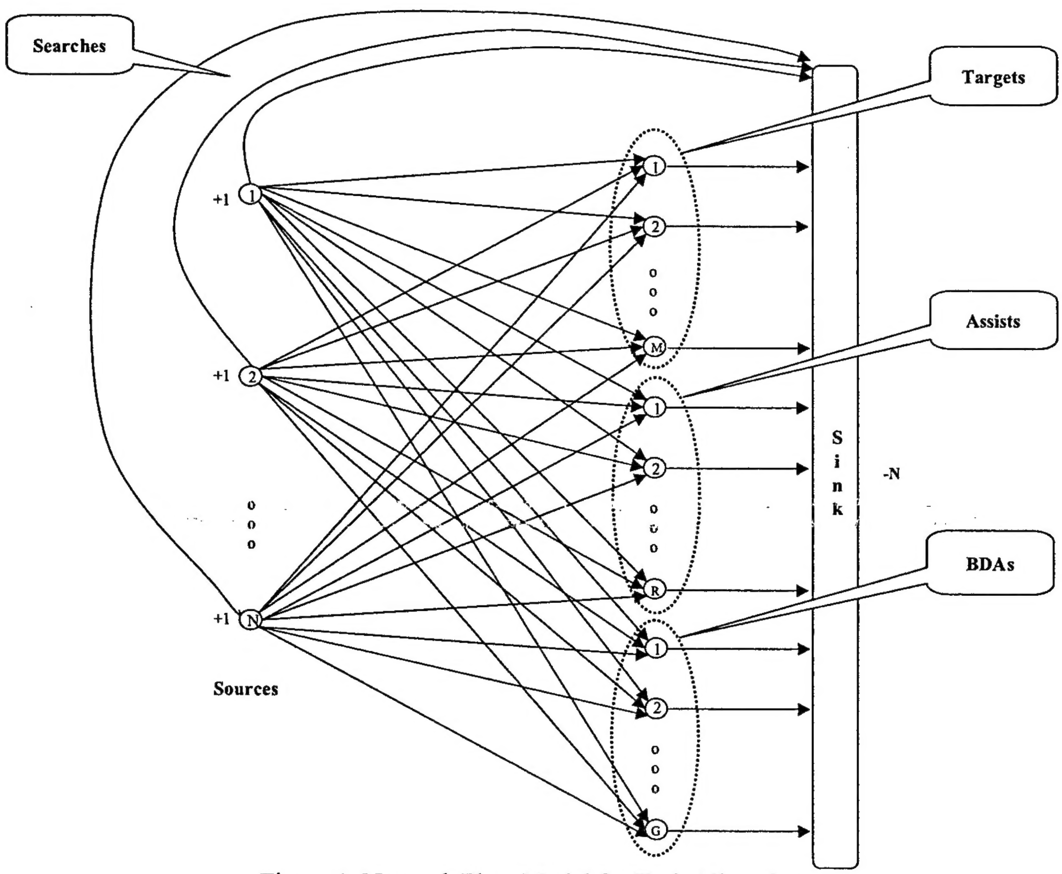

Figure 1: Network Flow Model for Task Allocation

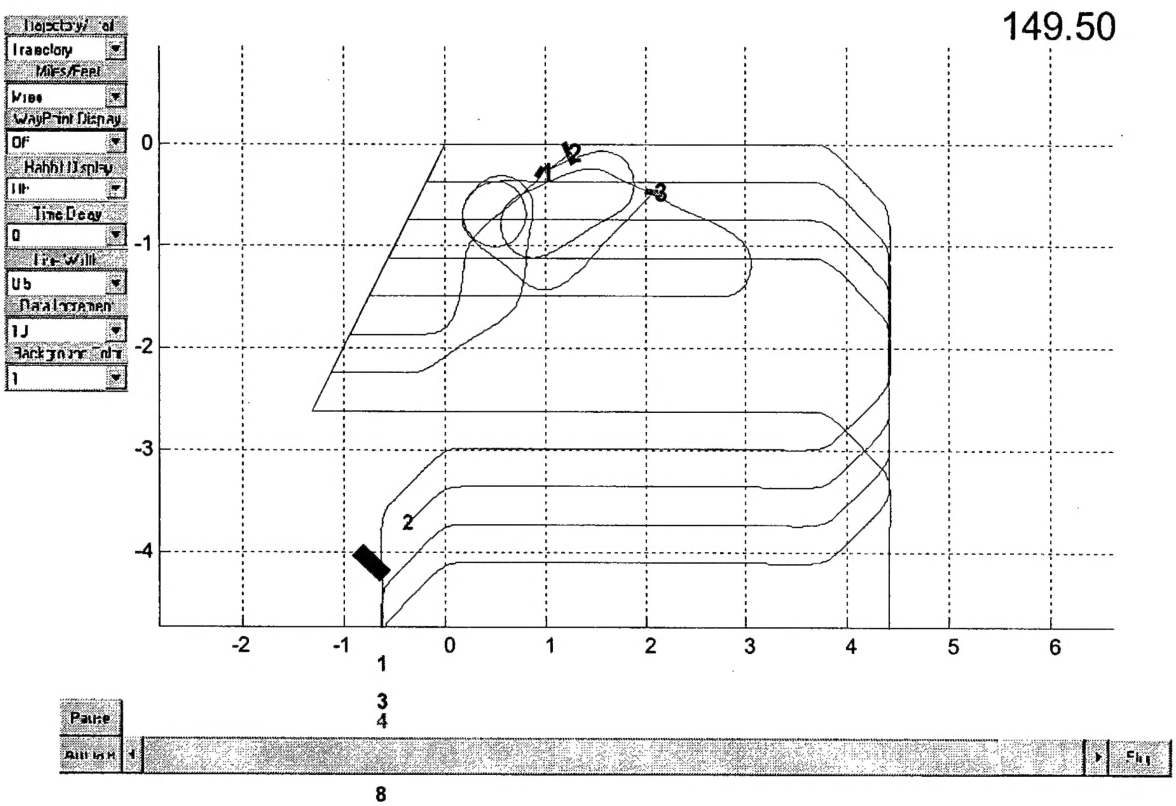

Figure 2: Vehicle Flight Paths and Target Locations